# Using Let's Encrypt with K8S Ingress Controllers

Will Plusnick
Developer Advocate
IBM


# Will Plusnick
## Developer Advocate for K8S at IBM

pwplusni@us.ibm.com
@WillPlusnick

# Topics Covered in This Talk

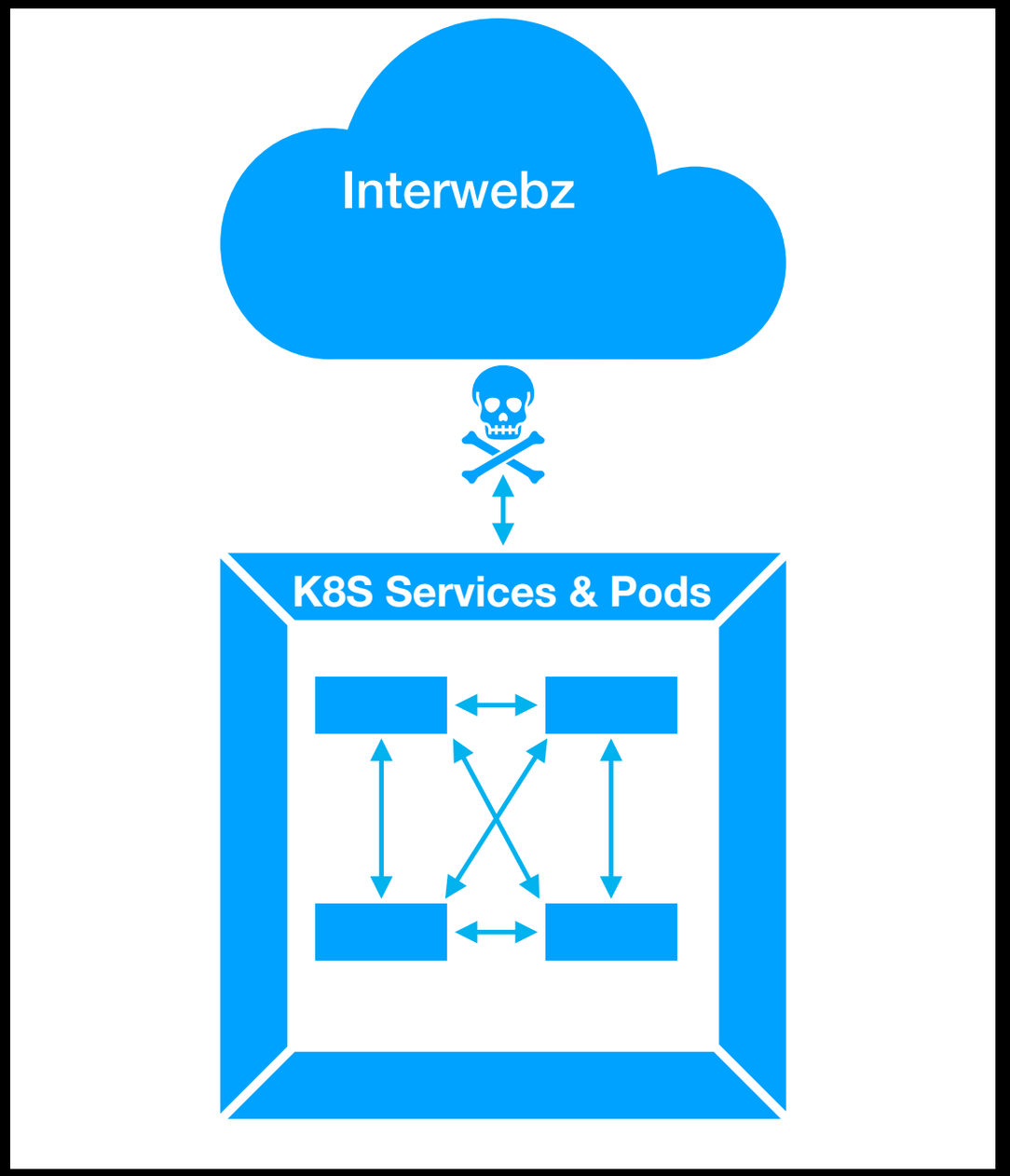
- K8S Ingress Controllers
    - What are K8S Ingress Controllers
    - Types of Ingress Controllers
- Let’s Encrypt
    - What is Let’s Encrypt
    - Why should you care about Let’s Encrypt
    - Intro into the ACME protocol
- Combining K8S Ingress Controllers
    - cert-manager
- Q&A

# What Will Not Be Covered

- K8S use in general
- Certificate authorities other than Let’s Encrypt
- Vendor specific issues
- Why Emacs is better than VIM (even though it clearly is)

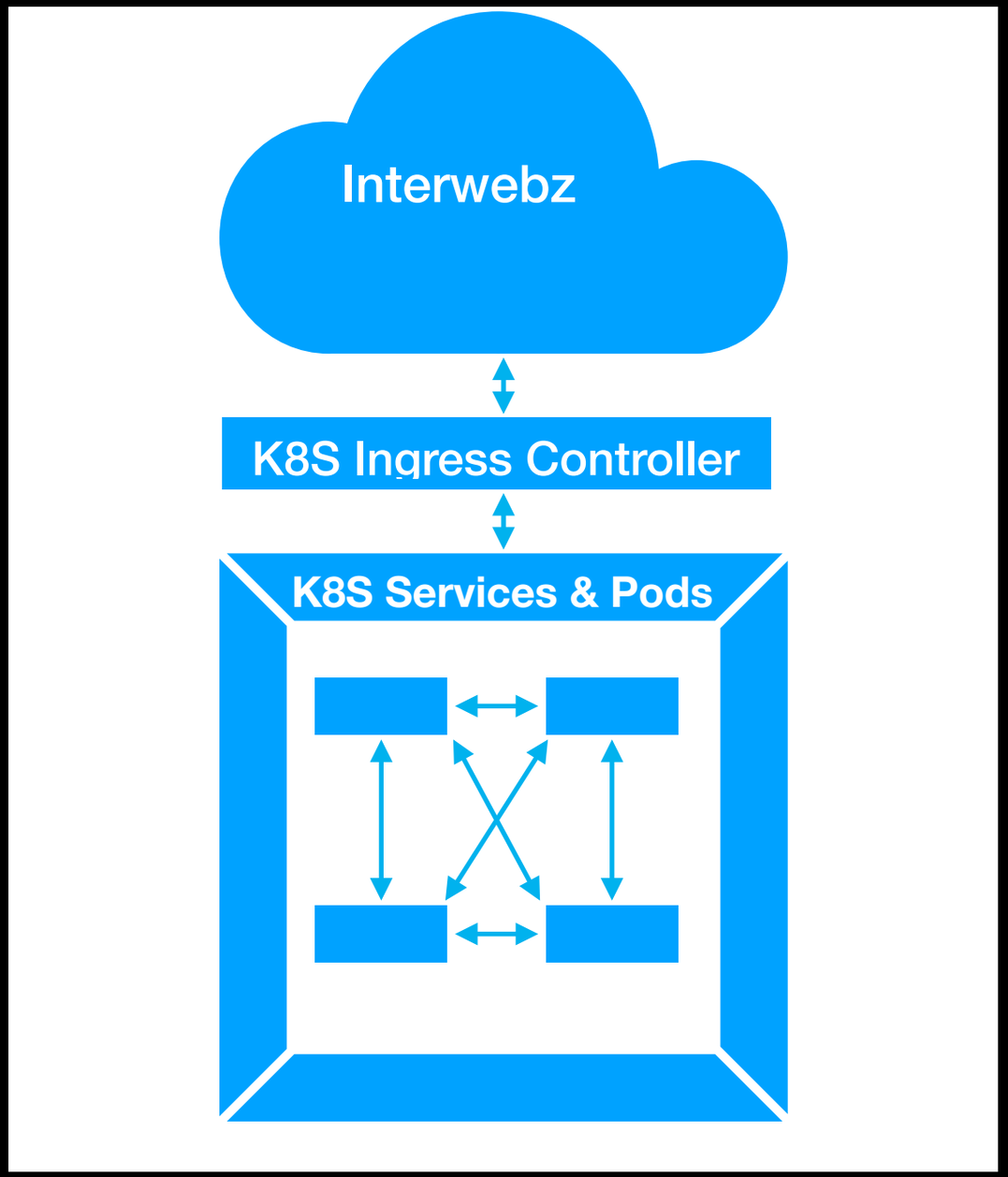
# What are K8S Ingress Controllers

- K8S pods and services are only available within the cluster
- So you’re application works and is secure, right?
- Only problem is an application needs users and customers



# What are K8S Ingress Controllers

- K8S ingress controllers bridge this gap
- They allow outside connections to reach the internal cluster network



# Ingress Controller Types

- Single Service
  - All requests go to a single k8s service
- Fan Out
  - Different paths are mapped to different services
- Virtual Host
  - Different subdomains are mapped to different services
- TLS
  - Certificate stored in Kubernetes secret

# What is Let’s Encrypt

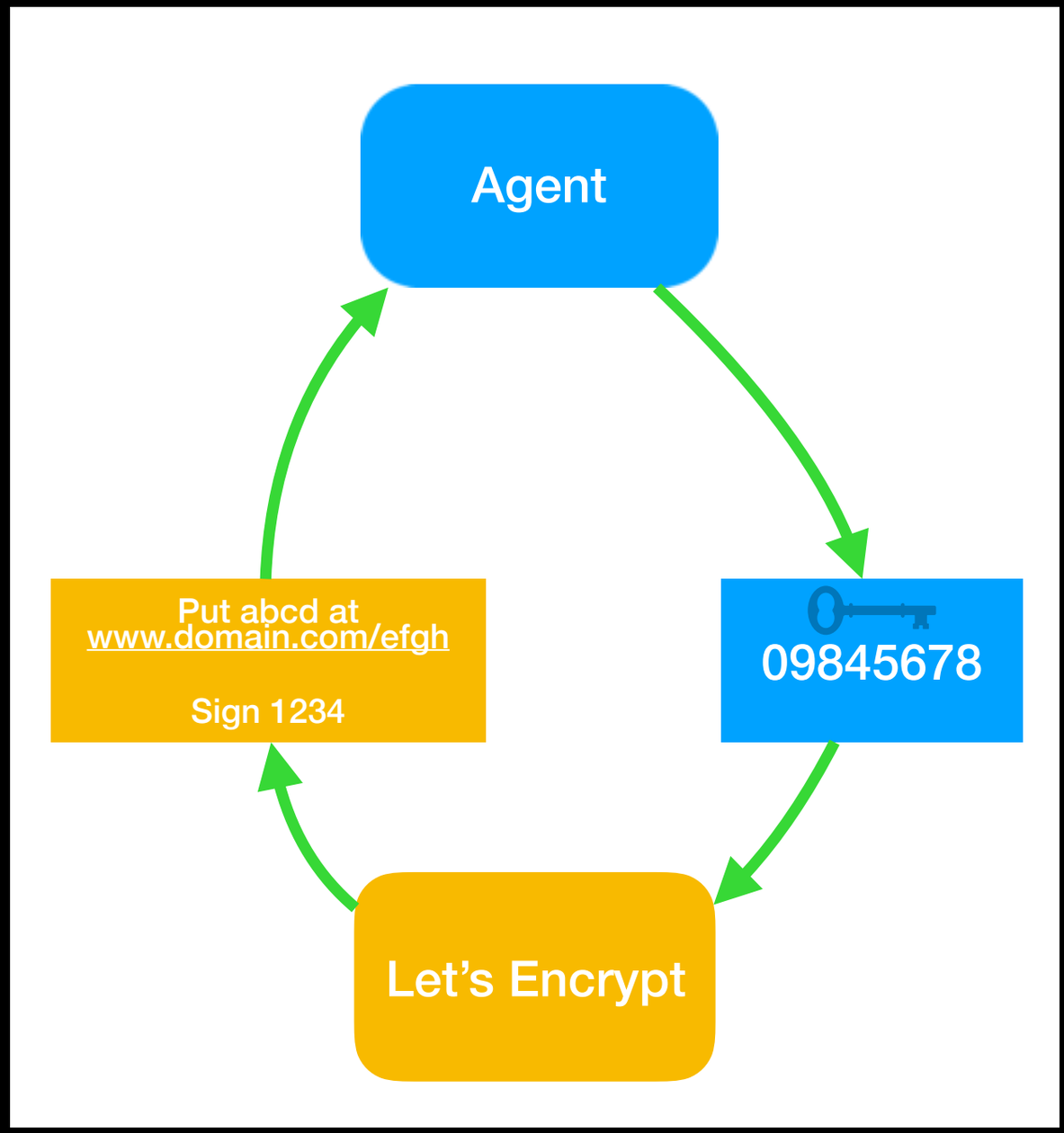
- Let’s Encrypt is a trusted Certificate Authority (CA)
- Certificates from a trusted CA used to costs a lot of money
- Let’s Encrypt is a free, automatic, and easy to use

# ACME Protocol

- Let’s Encrypt Certification’s Two phases
  - Establishing control of a domain
  - Requesting, revoking, and renewing certificates

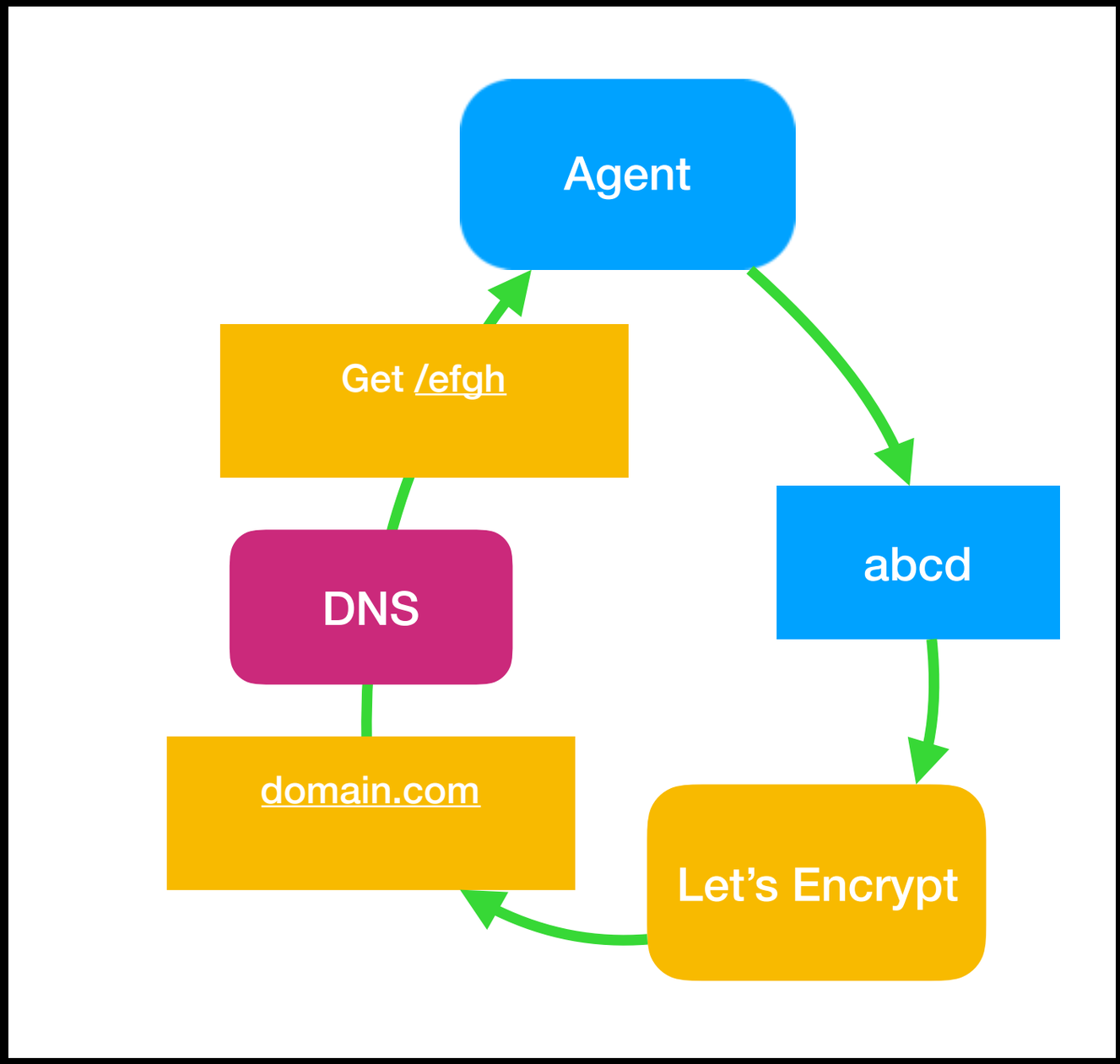
# ACME Protocol

- Agent creates a public-private key pair
- A challenge is issued to the agent
  - A nonce that is to be encrypted with the private key
  - An action done to or on the domain



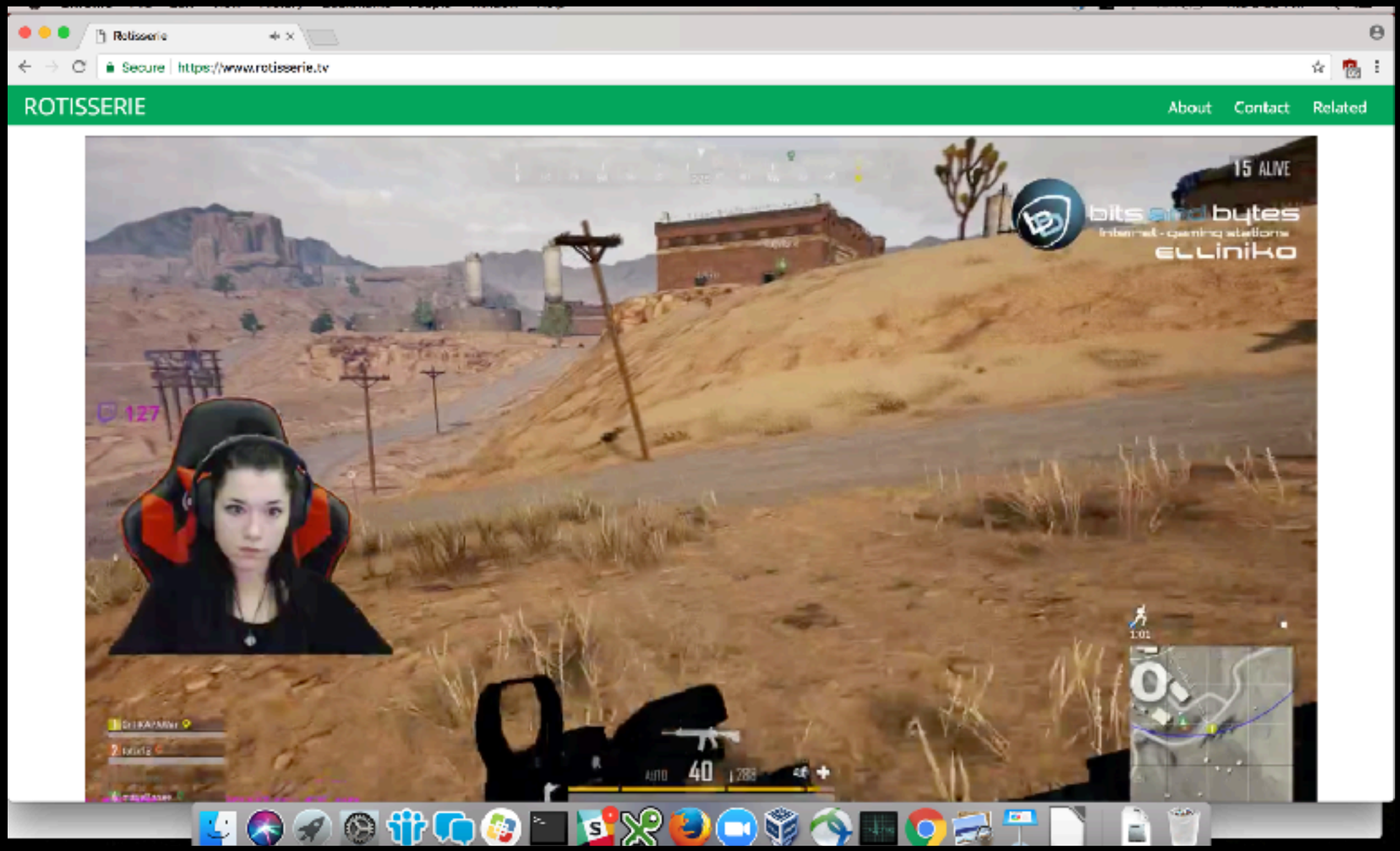
# ACME Protocol

- Let’s Encrypt now knows the server as the holder of the private key
- It checks the action requested was also done to the server
- Now it has established an identity of the server and that it controls the domain



# ACME Protocol

- Established
  - Authorized key pair
  - Server controls domain
- Authorized key pairs are able:
  - PKCS#10 Certificate Signing Request
  - Revoke certificates
  - Etc



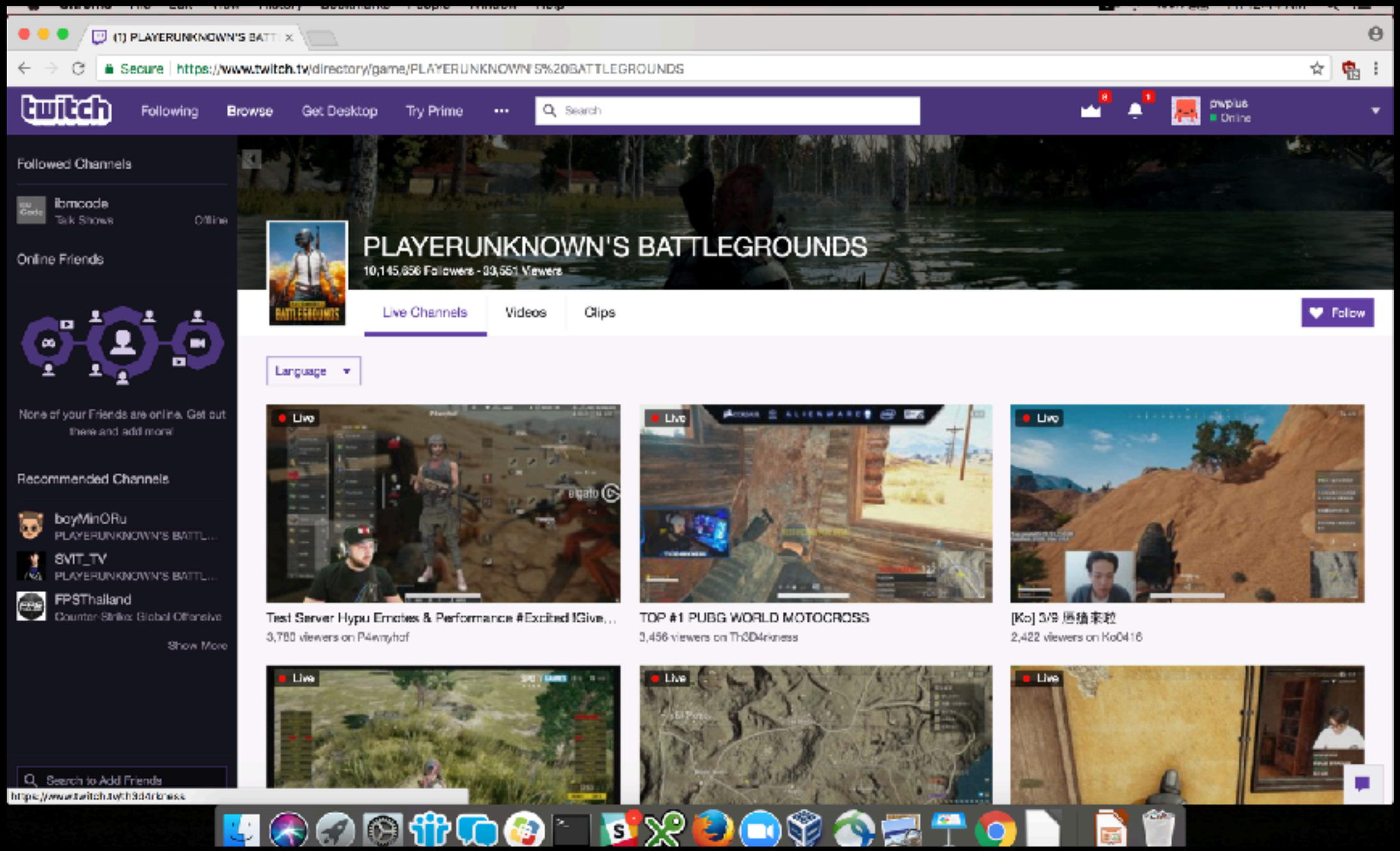

# rotisserie.tv

An open source https Kubernetes based application

# PUBG

- PlayerUnkown’s BattleGrounds
  - 100 person battle royale game
  - Dropped with no ammo, weapons, armor, etc
    - Last person (or team) left alive wins
- 20 Million Copies sold
- Not technically finished
- Slow paced
  - Games last 20-40 minutes
  - First 15 minutes are looting

# Twitch


(1) PLAYERUNKNOWN'S BATT
Secure
https://www.twitch.tv/directory/game/PLAYERUNKNOWN'S%20BATTLEGROUNDS
twitch
Following
Browse
Get Desktop
Try Prime
Search
Followed Channels
ibmcode
Talk Shows
Offline
Online Friends
None of your Friends are online. Get out there and add more!
Recommended Channels
boyMinORu
PLAYERUNKNOWN'S BATTL...
SMT_TV
PLAYERUNKNOWN'S BATTL...
FPSThailand
Counter-Strike: Global Offensive
Show More
Search to Add Friends
PLAYERUNKNOWN'S BATTLEGROUNDS
10,145,856 Followers - 93,561 Viewers
Live Channels
Videos
Clips
Follow
Language
Live
Test Server Hypu Emotes & Performance #Excited !Give...
3,760 viewers on P4wnyhof
TOP #1 PUBG WORLD MOTOCROSS
3,456 viewers on Th3D4rkness
[Ko] 3/9 愚搞來趕
2,422 viewers on Ko0416

# Livestreamer

- Cli application for streaming various videos including twitch
- Easy to use and just works

# TensorFlow Based OCR

- An open source deep learning framework
- Custom trained on the fonts

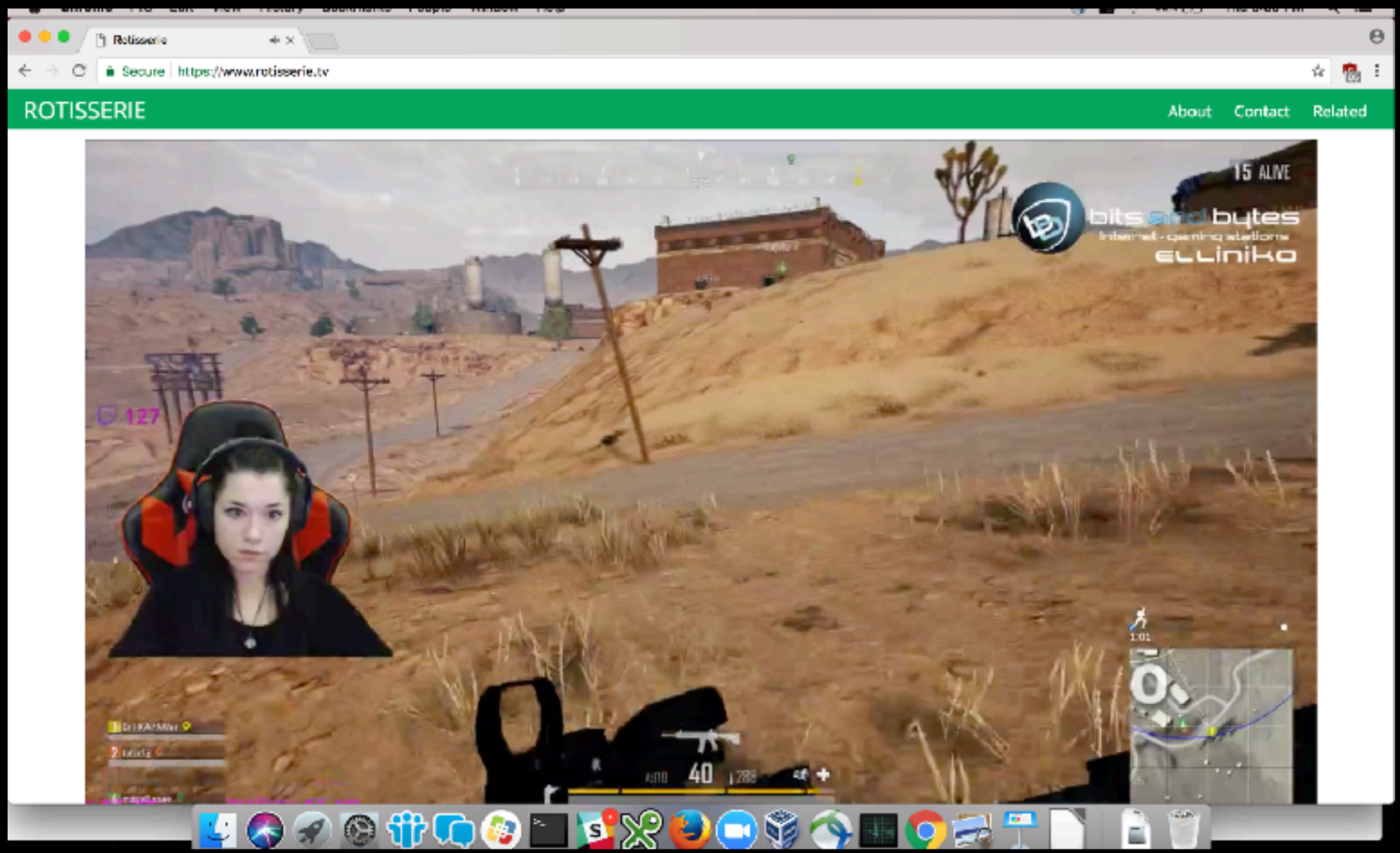
# FFMPEG

- A multimedia manipulation tool
- Has a CLI
- Open source

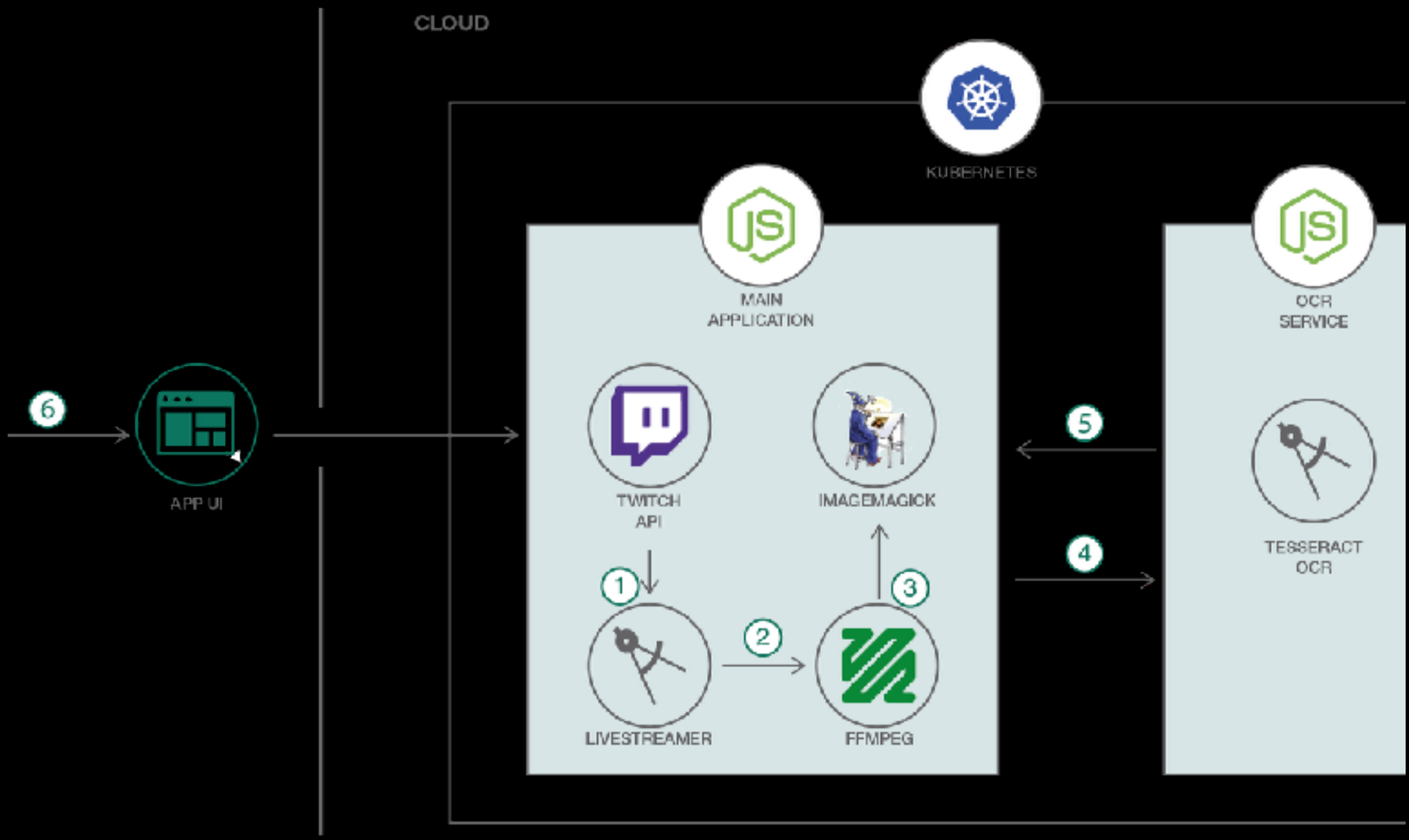
# ImageMagick

- Image manipulation suite
- Works on many different file formats
- Open source

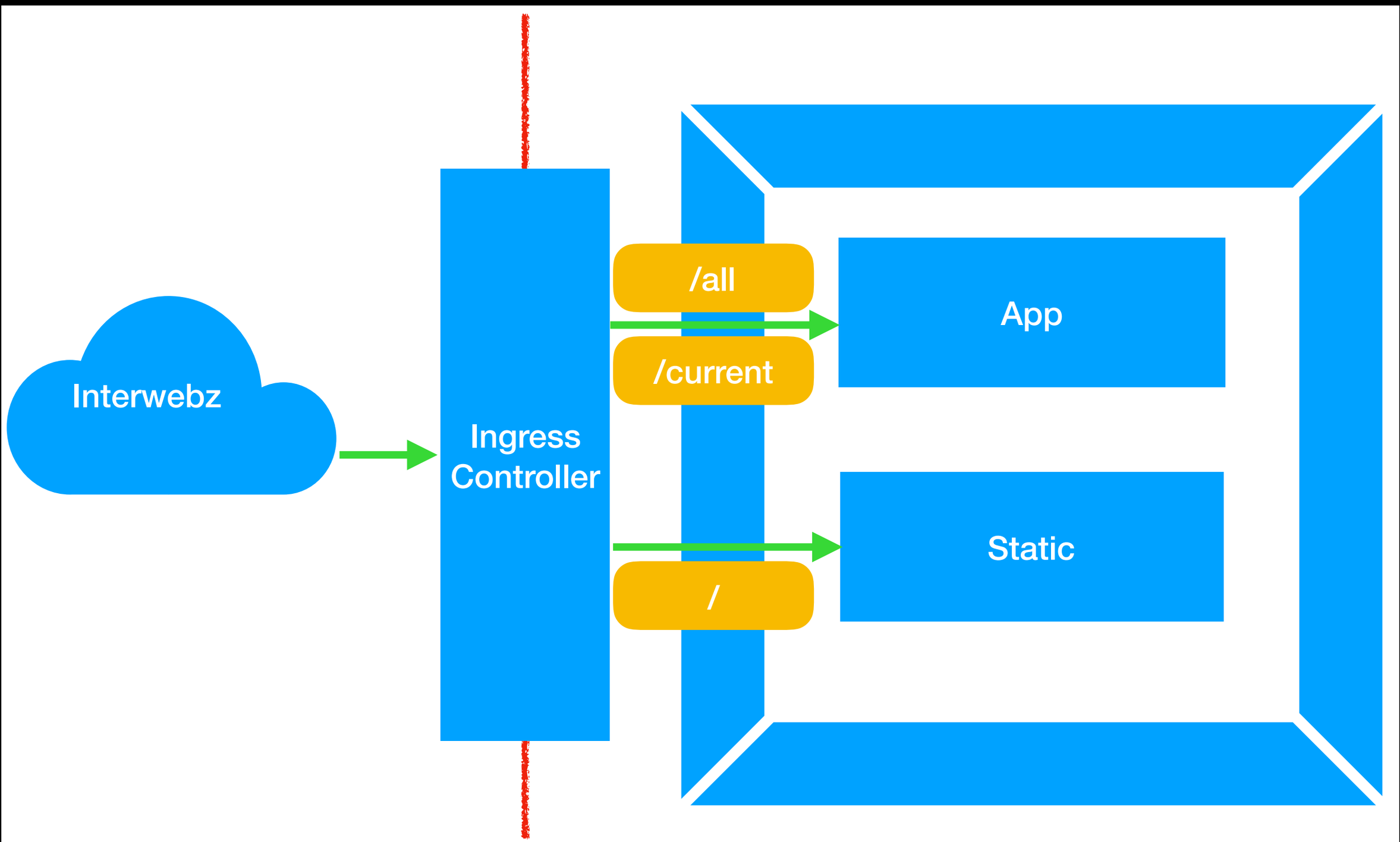
# Deciding the best stream



# rotisserie architecture



# rotisserie Ingress



# Nginx Ingress Controller

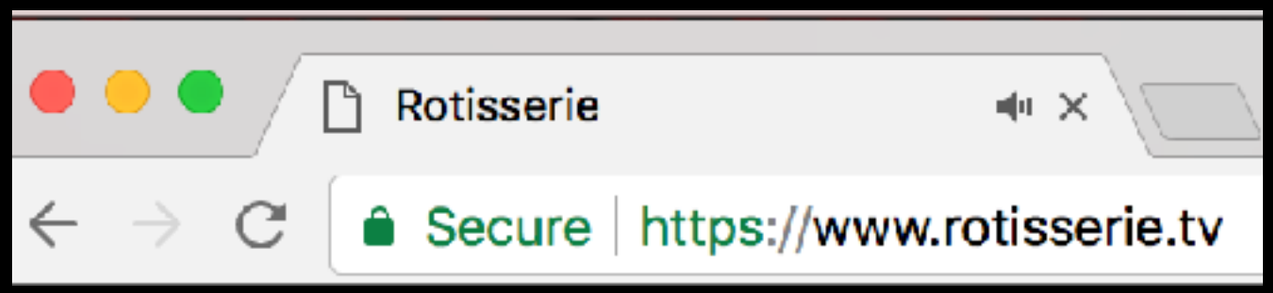
```
---
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: {{ template "rotisserie.fullname" . }}-ingress
  chart: {{ template "rotisserie.chart" . }}
  release: {{ .Release.Name }}
  heritage: {{ .Release.Service }}
  annotations:
    kubernetes.io/tls-acme: "True"
    ingress.bluemix.net/redirect-to-https: "True"
spec:
  tls:
  - secretName: {{ template "rotisserie.fullname" .}}-tls
    hosts:
  {{- range .Values.ingress.hostnames }}
    - {{ . }}
  {{- end }}
  rules:
  {{- range .Values.ingress.hostnames }}
  - host: {{ . }}
    http:
      paths:
      - path: /current
        backend:
          serviceName: {{ template "rotisserie.fullname" $ }}-app
          servicePort: 3000
      - path: /
        backend:
          serviceName: {{ template "rotisserie.fullname" $ }}-static
          servicePort: 8082
```

# cert-manager

- sets up service that automates certificate actions

# Demo

# Finally




- Working application
- Https enabled
- Automatically keeps up-to-date

**https://rotisserie.tv**

# Resources

- Ingress Controllers — https://kubernetes.io/docs/concepts/services-networking/ingress/
- Let's Encrypt — https://letsencrypt.org/
- cert-manager —https://github.com/jetstack/cert-manager/
- kube-lego to cert-manager migration guide — https://github.com/jetstack/cert-manager/blob/master/docs/user-guides/migrating-from-kube-lego.md
- rotisserie — rotisserie.tv and github.com/ibm/rotisserie

# Questions?